The \texttt{bussproofs-extra} package\footnote{This document corresponds to \texttt{bussproofs-extra} 0.4, dated 2019/05/31.}

Richard Zach

1 Introduction

The \texttt{bussproofs-extra} package provides additional functionality for the proof tree typesetting package \texttt{bussproofs} by Sam Buss. It is experimental and tested only with v.1.1, and only in L\TeX\ mode with upward-growing trees. Functionality provided includes:

1. \texttt{\textbackslash Deduce$} and \texttt{\textbackslash DeduceC} commands, which work much like \texttt{\textbackslash Infer} commands but indicate missing parts of a proof.

2. Multiple styles for typesetting the result of \texttt{\textbackslash Deduce}, including
   
   (a) \texttt{\textbackslash straightDeduce}, which produces vertical dots
   
   (b) \texttt{\textbackslash branchDeduce}, which produces diagonal plus vertical dots
   
   (c) \texttt{\textbackslash dotsDeduce}, which produces diagonal dots from top left to bottom right
   
   (d) \texttt{\textbackslash dotsdDeduce}, which produces diagonal dots from top right to bottom left
   
   (e) \texttt{\textbackslash shortDeduce}, which is like \texttt{\textbackslash straightDeduce} but half the length

   \texttt{\textbackslash straightDeduce} is the default. It can be changed by redefining \texttt{\textbackslash alwaysDeduce}.

3. \texttt{\textbackslash LeftLineLabel} and \texttt{\textbackslash RightLineLabel} commands which work like \texttt{\textbackslash LeftLabel} and \texttt{\textbackslash RightLabel} but place a label next to the conclusion of an inference/deduction instead of the score line.

4. \texttt{\textbackslash LeftSubproofLabel} and \texttt{\textbackslash RightSubproofLabel} commands which work like \texttt{\textbackslash LeftLabel} and \texttt{\textbackslash RightLabel} but place a label next to the entire preceding subproof with a curly brace.

Here’s what these deductions look like:
The most up-to-date version of this package is available at the Open Logic Project github site, where you can file bug reports as well.

1.1 Example

```
\begin{prooftree}
  \AxiomC{}
  \RightLabel{$\pi_1(a)$}
  \Deduce\Gamma_1 \fCenter \Theta_1, F(a)$
  \RightLabel{$\forall$R}
  \UnaryInf\Gamma_1 \fCenter \Theta_1, \forall x\,F(x)$
  \ddotsDeduce
  \RightLabel{$\pi_1'$}
  \Deduce\Gamma \fCenter \Theta, \forall x\,F(x)$
  \AxiomC{}
  \RightLabel{$\pi_2$}
  \Deduce$F(n), \Delta_1 \fCenter \Lambda_1$
  \RightLabel{$\forall$L}
  \UnaryInf$\forall x\,F(x), \Delta_1 \fCenter \Lambda_1$
  \dotsdDeduce
  \RightLabel{$\pi_2'$}
  \Deduce$\forall x\,F(x), \Delta \fCenter \Lambda$
  \RightLabel{cut}
  \BinaryInf$\Gamma, \Delta \fCenter \Theta, \Lambda$
  \RightLabel{$\pi_4$}
  \branchDeduce
  \Deduce$\Pi \fCenter \Xi$
\end{prooftree}
```
It is also possible to label entire subproofs on the left and on the right.

\begin{prooftree}
\AxiomC{}
\Deduce\Gamma \fCenter \Delta\LeftLineLabel{$S_1$}
\Deduce\Gamma \fCenter \Delta, A\LeftSubproofLabel{$\pi$}
\AxiomC{}
\Deduce\Gamma' \fCenter \Delta'\LeftLineLabel{$S_2$}
\Deduce A, \Gamma' \fCenter \Delta'\RightSubproofLabel{$\pi'$}
\RightLabel{cut}\LeftLineLabel{$S_3$}
\BinaryInf\Gamma, \Gamma' \fCenter \Delta, \Delta'\Deduce\Pi \fCenter \Lambda\end{prooftree}

The \texttt{\LeftLineLabel} and \texttt{\RightLineLabel} commands add labels to the sequent or formula produced by the following \texttt{\Axiom}, \texttt{\XxxxInf}, and \texttt{\Deduce} command.
\begin{prooftree}
  \LeftLineLabel{$S_1$}
  \RightLineLabel{$S_1$}
  \Axiom$\phantom{A, {}} \Gamma \fCenter \Delta$
  \LeftLineLabel{$S_2$}
  \RightLineLabel{$S_2$}
  \UnaryInf$A, \Gamma \fCenter \Delta, B$
  \LeftLineLabel{$S_3$}
  \Deduce$\Pi \fCenter \Lambda$
  \LeftLineLabel{$S_1'$}
  \AxiomC{$A$}
  \LeftLineLabel{$S_2'$}
  \UnaryInfC{$A \lor B$}
  \LeftLineLabel{$S_3'$}
  \DeduceC{$C$}
  \LeftLineLabel{$S_4$}
  \BinaryInf$\Pi \fCenter \Lambda$
  \RightLineLabel{$S_5$}
  \UnaryInf$\Pi \fCenter \Lambda$
\end{prooftree}

If the sequent or formula is itself a premise of an \texttt{\textbackslash{XxxInf}} command and the conclusion is longer, this may produce a less than optimal result, as the label is produced before the score line below (compare the left and right labels of the top left sequent above). In that case you may want to insert extra space using \texttt{\textbackslash{phantom}}, or use \texttt{\textbackslash{makebox}} and the \texttt{\textbackslash{widthof}} command of the \texttt{calc} package for the \texttt{XxxC} variants of the commands (see the top right formula below) as in the \texttt{\textbackslash{Axiom}} commands below.

\begin{verbatim}
\begin{prooftree}
  \LeftLineLabel{$S_1$}
  \RightLineLabel{$S_1$}
  \Axiom$\phantom{A, {}} \Gamma \fCenter \Delta$
  \LeftLineLabel{$S_2$}
  \RightLineLabel{$S_2$}
  \UnaryInf$A, \Gamma \fCenter \Delta, B$
  \LeftLineLabel{$S_3$}
  \Deduce$\Pi \fCenter \Lambda$
  \LeftLineLabel{$S_1'$}
  \AxiomC{$A$}
  \LeftLineLabel{$S_2'$}
  \UnaryInfC{$A \lor B$}
  \LeftLineLabel{$S_3'$}
  \DeduceC{$C$}
  \LeftLineLabel{$S_4$}
  \BinaryInf$\Pi \fCenter \Lambda$
  \RightLineLabel{$S_5$}
  \UnaryInf$\Pi \fCenter \Lambda$
\end{prooftree}
\end{verbatim}

\section{Implementation}

\subsection{Setup}

We require \texttt{bussproofs} (obviously) and and \texttt{tikz} for drawing things.

1 \texttt{\textbackslash{RequirePackage}{bussproofs}}
2 \texttt{\textbackslash{RequirePackage}{tikz}}
2.2 Dimensions and boxes

bussproofs aligns sequents at the right end of the sequent arrow, so we need to remember by how much to correct to get deductions to the middle of sequents. For \ddotsDeduce and \dotsdDeduce (diagonal) styles, the upper and lower sequents will be displaced.
\newdimen\CenterCorrection
\newdimen\DiagCorrection

We need two boxes to hold the left and right line labels.
\newbox\myBoxLLL
\newbox\myBoxRLL

2.3 Deduce Styles

The following commands set the style for the next \Deduce command. \straightDeduce produces a simple vertical line of dots, and \shortDeduce a line of half that length. \branchDeduce produces centered branching (Takeuti/Gentzen-style) dots, \ddotsDeduce left-to-right diagonal dots, and \dotsdDeduce right-to-left diagonal dots. They do this by redefining the \fDeduce command which produces the dots and sets up the dimensions. The TikZ style deduceLine is used as argument to the \draw command and can be redefined for other line styles as well (e.g., smaller dots or closer spacing).
\tikzset{
  deduceLine/.style = {line width=1.1pt, loosely dotted}}
\def\straightDeduce{%
  \gdef\fDeduce{\tikz\draw[deduceLine] (0,0) -- (0,1);}
  \global\DiagCorrection=0pt
  \ignorespaces}
\def\shortDeduce{%
  \gdef\fDeduce{\tikz\draw[deduceLine] (0,0) -- (0,.5);}
  \global\DiagCorrection=0pt
  \ignorespaces}
\def\branchDeduce{%
  \gdef\fDeduce{\begin{tikzpicture}
  \draw[deduceLine] (0,0) -- (0,1);
  \draw[deduceLine] (-.5,.5) -- (0,0);
  \draw[deduceLine] (.5,.5) -- (0,0);
  \end{tikzpicture}}
  \global\DiagCorrection=0pt
  \ignorespaces}
\def\ddotsDeduce{%
  \gdef\fDeduce{\begin{tikzpicture}
  \draw[deduceLine] (0,0) -- (0,1);
The \texttt{alwaysDeduce} command is used to (re)set the deduce style to a default and is executed every time a deduction is typeset. It can be redefined to change the default deduce style.

\texttt{\def\alwaysDeduce{\straightDeduce}}

\texttt{\straightDeduce}

\textbf{2.4 \texttt{\Deduce$} and \texttt{\DeduceC}}

\texttt{\Deduce$} and \texttt{\DeduceC} are the commands to actually produce the deductions. They are used and work just like \texttt{\UnaryInf$} and \texttt{\UnaryInfC}.

\texttt{\def\Deduce$#1\fCenter#2${%}
\texttt{\prepUnary%}
\texttt{\buildConclusion{#1}{#2}%}
\texttt{\setbox\myBoxA=hbox{\fCenter}%}
\texttt{\global\CenterCorrection=-.5\wd\myBoxA}
\texttt{\joinDeduce%}
\texttt{\resetInferenceDefaults%}
\texttt{\ignorespaces%}
}\texttt{}}

\texttt{\def\DeduceC#1{%}
\texttt{\prepUnary%}
\texttt{\buildConclusionC{#1}%}
\texttt{\setbox\myBoxA=hbox{\fCenter}%}
\texttt{\global\CenterCorrection=0pt}
\texttt{\joinDeduce%}
\texttt{\resetInferenceDefaults%}
\texttt{\ignorespaces%}
\texttt{}}}
3 Typesetting the Deduction

\joinDeduce aligns and joins \curBox and \myBoxC into a single vbox. \curBox holds the upper proof, \curScoreStart is distance to where the line below the premise would start, \curScoreCenter is distance from left edge of score to the alignment point, and \curScoreEnd is width of the score line.

\def\joinDeduce{%
  \global\advance\curCenter by -\hypKernAmt%

  If center of premise is left of center of conclusion move upper box to right by difference, else move lower box right by difference

  \ifnum\curCenter<\newCenter%
    \displace=\newCenter%
    \advance \displace by -\curCenter%
    \kernUpperBox%
  \else%
    \displace=\curCenter%
    \advance \displace by -\newCenter%
    \kernLowerBox%
  \fi%

  For \ddotsDeduce, move lower box right; for \dotsdDeduce, move upper box right; then set \curCenter to align with horizontal center of dots.

  \ifnum\DiagCorrection<0%
    \displace=-\DiagCorrection
    \kernLowerBox%
  \else
    \displace=\DiagCorrection
    \kernUpperBox%
  \fi%

  \advance\curCenter by-.5\DiagCorrection

Now we draw the deduction.

\buildDeduce%

Put the deduction and labels into a box.

\buildScoreLabels%

Put everything into a new box and compute the dimensions for the next \Deduce or \XxxxInf.

\ifx\rootAtBottomFlag\myTrue%
  \buildRootBottom%
\else%
  \buildRootTop%
\fi%
\buildDeduce does for \DeduceX what \buildInf does for \XxxInf: put the
deduction bit (dots) into a box and set the dimensions properly.

\begin{verbatim}
def\buildDeduce{\global\setbox\myBoxD equals\hbox{\fDeduce}\displace = \wd\myBoxD \% find width of vdots
set start and end of current score to left and right of the box holding the deduction.
\global\curScoreStart = \curCenter\global\advance\curScoreStart by -.5\displace\global\curScoreEnd = \curCenter\global\advance\curScoreEnd by .5\displace\global\advance\curScoreStart by\CenterCorrection
\global\advance\curScoreEnd by\CenterCorrection}
\end{verbatim}

3.1 Line Labels

\LeftLineLabel and \RightLineLabel set the label to place to the left or right,
respectively, of the conclusion of the next \Axiom, \XxxInf or \Deduce command.
They are aligned with the text produced by \LeftLabel and \RightLabel (i.e.,
the distance to the line is \ScoreOverhang + \labelSpacing).

\begin{verbatim}
def\LeftLineLabel#1{\global\def\displayLeftLineLabel{#1\hskip\labelSpacing}}\global\let\displayLeftLineLabel\relax
def\RightLineLabel#1{\global\def\displayRightLineLabel{\hskip\labelSpacing #1}}\global\let\displayRightLineLabel\relax
\end{verbatim}

3.2 Subproof Labels

Sometimes you’d like to label entire subproofs. This is done with commands
\LeftSubproofLabel and \RightSubproofLabel.

\begin{verbatim}
def\LeftSubproofLabel#1{\global\setbox\curBox =\hbox{\vbox to \ht\curBox{\vfil\llap{#1$\left\{\vrule height .5\ht\curBox width 0pt\right.$}}\vfil}}\global\let\displayLeftSubproofLabel\relax\global\let\displayRightSubproofLabel\relax
\end{verbatim}
3.3 Patched commands from bussproofs

Some commands from \texttt{bussproofs.sty} have to be redefined to include \texttt{bussproofs-extra} functionality. Added/changed lines are indicated by a \%\texttt{bpextra} comment.

\begin{verbatim}
\def\resetInferenceDefaults{%
  \global\def\theHypSeparation{\defaultHypSeparation}%
  \global\setbox\myBoxLL=\hbox{\defaultLeftLabel}%
  \global\setbox\myBoxRL=\hbox{\defaultRightLabel}%
  \global\def\buildScore{\alwaysBuildScore}%
  \global\def\theScoreFiller{\alwaysScoreFiller}%
  \% reset line labels to nothing \%bpextra
  \global\let\displayLeftLineLabel\relax \%bpextra
  \global\let\displayRightLineLabel\relax \%bpextra
  \% reset to default deduce style \%bpextra
  \alwaysDeduce \%bpextra
  \let\hypKernAmt0pt% Restore to zero kerning.
  \}

\def\Axiom$#1\fCenter#2${%
  \prepAxiom%
  \% Get level and correct names set.
  \%bpextra -- add line labels
  \setbox\myBoxA=\hbox{$\mathord{#1}\fCenter\mathord{\relax}$}%
  \setbox\myBoxB=\hbox{$#2$}%
  \setbox\myBoxLLL=\hbox{\displayLeftLineLabel}%
  \setbox\myBoxRLL=\hbox{\displayRightLineLabel}%
  \setbox\myBoxLLL=}\hbox{\displayLeftLineLabel}%
  \setbox\myBoxRLL=}\hbox{\displayRightLineLabel}%
  \setbox\curBox=%
  \unhcopy\myBoxLLL%bpextra
  \hskip\ScoreOverhangLeft\relax
  \unhcopy\myBoxA
  \unhcopy\myBoxB
  \hskip\ScoreOverhangRight
  \unhcopy\myBoxRLL%bpextra
  \% Set the relevant dimensions for the boxes
  \global\curScoreStart=0pt \relax
  \global\curScoreEnd=\wd\curBox \relax
  \global\curCenter=\wd\myBoxA \relax \%bpextra
\end{verbatim}
\global\advance \curCenter by \ScoreOverhangLeft\%  
% bpextra adjust by dimensions of labels
\global\advance \curCenter by \wd\myBoxLLL\%bpextra
\global\advance \curScoreStart by \wd\myBoxLLL\%bpextra
\global\advance \curScoreEnd by -\wd\myBoxRLL\%bpextra
% reset line labels to nothing %bpextra
\global\let\displayLeftLineLabel\relax \%bpextra
\global\let\displayRightLineLabel\relax \%bpextra
\ignorespaces
}

\def\AxiomC#1{ % Note argument not in math mode
% Get level and correct names set.
\prepAxiom%
% Define the box.
\setbox\myBoxA=\hbox{#1}%
\setbox\myBoxLLL=\hbox{\displayLeftLineLabel}% %bpextra
\setbox\myBoxRLL=\hbox{\displayRightLineLabel}% %bpextra
\global\setbox\curBox =%
\hbox{\unhcopy\myBoxLLL\%bpextra
  \hskip\ScoreOverhangLeft\relax%
  \unhcopy\myBoxA
  \hskip\ScoreOverhangRight\relax
  \unhcopy\myBoxRLL}\% \bpextra
\ignorespaces
% Set the relevant dimensions for the boxes
\global\curScoreStart=0pt \relax
\global\curScoreEnd=\wd\curBox \relax
\global\curCenter=.5\wd\myBoxA \relax \%bpextra
\global\advance \curCenter by \ScoreOverhangLeft\%
% bpextra adjust by dimensions of labels
\global\advance \curCenter by \wd\myBoxLLL\%bpextra
\global\advance \curScoreStart by \wd\myBoxLLL\%bpextra
\global\advance \curScoreEnd by -\wd\myBoxRLL\%bpextra
% reset line labels to nothing \%bpextra
\global\let\displayLeftLineLabel\relax \%bpextra
\global\let\displayRightLineLabel\relax \%bpextra
\ignorespaces
}

\def\buildConclusion#1#2{% Build lower sequent w/ center at \fCenter position.
% Define the boxes
\setbox\myBoxA=\hbox{$\mathord{#1}\fCenter\mathord{\relax}$}%
\setbox\myBoxB=\hbox{$#2$}%
\setbox\myBoxLLL=\hbox{\displayLeftLineLabel}% \%bpextra
\setbox\myBoxRLL=\hbox{\displayRightLineLabel}% \%bpextra
\setbox\myBoxC =%
\hbox{\unhcopy\myBoxLLL\%bpextra
  \hskip\ScoreOverhangLeft\relax%
  \unhcopy\myBoxA
  \unhcopy\myBoxB
  \hskip\ScoreOverhangRight\relax
  \unhcopy\myBoxA
  \unhcopy\myBoxB}%
\hspace{\ScoreOverhangRight}
 unhcopy\myBoxRLL)% %bpextra
 % Calculate the center of the \myBoxC string.
 \newScoreStart=0pt \relax%
 \newCenter=\wd\myBoxA \relax%
 \advance \newCenter by \ScoreOverhangLeft%
 \newScoreEnd=\wd\myBoxC%
 % bpextra adjust by dimensions of labels
 \global\advance\newCenter by \wd\myBoxLLL%bpextra
 \global\advance\newScoreStart by \wd\myBoxLLL%bpextra
 \global\advance\newScoreEnd by -\wd\myBoxRLL%bpextra
 \% reset line labels to nothing %bpextra
 \global\let\displayLeftLineLabel\relax %bpextra
 \global\let\displayRightLineLabel\relax %bpextra
 }
 \def\buildConclusionC#1{% Build lower sequent w/o \fCenter present.
 % Define the box.
 \setbox\myBoxA=\hbox{#1}%
 \setbox\myBoxLLL=\hbox{\displayLeftLineLabel}% %bpextra
 \setbox\myBoxRLL=\hbox{\displayRightLineLabel}% %bpextra
 \setbox\myBoxC=%
 \hbox{unhcopy\myBoxLLL%bpextra
 \hskip\ScoreOverhangLeft\relax%
 unhcopy\myBoxA
 \hskip\ScoreOverhangRight
 unhcopy\myBoxRLL}%bpextra
 % Calculate kerning to line up centers
 \newScoreStart=0pt \relax%
 \newCenter=.5\wd\myBoxA \relax% bpextra
 \newScoreEnd=\wd\myBoxC%
 \advance \newCenter by \ScoreOverhangLeft%
 % bpextra adjust by dimensions of labels
 \global\advance\newCenter by \wd\myBoxLLL%bpextra
 \global\advance\newScoreStart by \wd\myBoxLLL%bpextra
 \global\advance\newScoreEnd by -\wd\myBoxRLL%bpextra
 % reset line labels to nothing %bpextra
 \global\let\displayLeftLineLabel\relax %bpextra
 \global\let\displayRightLineLabel\relax %bpextra
 }

Change History

v0.1 dotsdDeduce, added examples . 1
General: Initial version with deduce, linelabel functionality . 1 v0.3
v0.2 General: Rename to bussproofs-extra.sty . . . . . . . . 1
v0.4

General: Better implementation of line labels; add shortDeduce;

style deduce lines . . . . . 1